Homework: Crawling and Deduplication of Weapons Images Using Nutch and Tika
Due: October 15, 2015 12pm PT

---

1. Overview

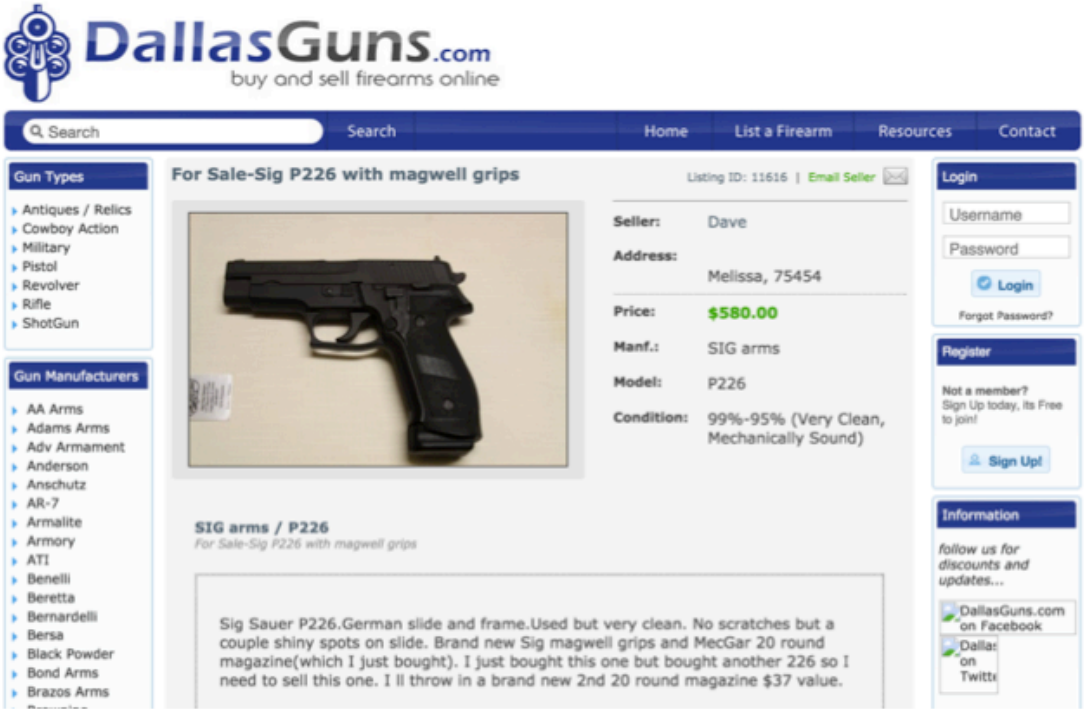

Figure 1: An example of a weapons classifieds site.

In this first assignment, we will emphasize techniques learned in class related to crawling, deduplication, similarity, and the vector space retrieval models. You will leverage Apache Nutch, Apache Tika (and its Python library), along with additional technologies including the Tika Similarity library to help identify images of weapons on the Internet. Weapons are frequently (legally) sold and traded on the Internet, but in many cases, depending on state, locality, weapon type, and purpose, the weapons may be nefariously traded and described on Internet forum sites. There is great interest in collecting a large corpus (100s of K – 1Ms) of images of these weapons, and automatically extracting information from them, and identifying which are duplicates, and which are similar to one another. This is an extremely topical issue especially considering gun violence in the U.S.

Of course, there are immense challenges in constructing such a corpus. First off, accessing these gun images, and gun sites may require the crawler to understand Ajax, and to point/click on links and pagination and logins as if it were a human browsing the Internet with a browser. Second, filtering for only images, when some or all of the image URLs don't necessary exhibit meaningful extension patterns (e.g., not all URLs end

in .png or .jpg, etc.) presents a challenge in filtering and selecting the appropriate URLs to crawl and fetch. Third, browsing through this data and its nature creates an impetus to provide mechanisms for a crawler to reduce its overall *footprint* that is its politeness levels should be high; and the crawler should have the ability to increase its sensitivity to the way that it appears in web logs for servers providing this weapons oriented content on the Internet. Finally, identifying unique images, and comparing multimedia data types is non-trivial, and even though there have been tremendous improvements in both accuracy, computational speed and memory requirements and so forth for image comparison algorithms, due to the rich metadata available from multimedia content and our ability to extract it, our assignment will focus on metadata based similarity approaches for deduplication and comparison.

Our goals are to collect between 500,000 to 1,000,000 unique images of weapons available on the Internet from the initial seed list. Then, our goals are to consider some thoughtful questions such as which weapons images are duplicated between different sites, and why, and then to consider if there are mechanisms to classify and compare weapons based on region, site type, and other information available from our crawled data such as price, and its affect overall on duplication and relationship of weapons across sites. We will also answer a series of challenge questions in this assignment being asked by leaders in the area of search engines and information retrieval and by members of the team working on the DARPA Memex effort, a national project to improve domain specific search.

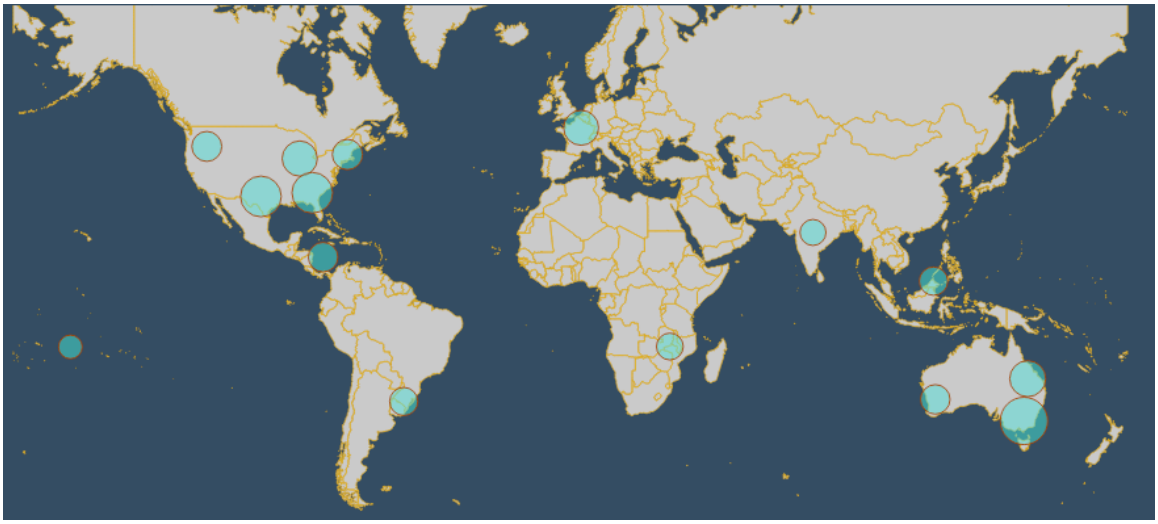

Figure 2: Density and location of weapons classifieds ads on the Internet.

2. Objective

The objective of this assignment is to collect *as many unique images of weapons available on the provided seed lists as possible*. At its core this will involve using the Apache Nutch (http://nutch.apache.org/) framework and its configurable web scale crawler to find and locate these images, and to during, and after the crawling process, to use the Apache Tika (http://tika.apache.org/) framework to perform analysis and analytics

activities including deduplication. You will extend Nutch and its Selenium Protocol Plugin (originally developed with help from CSCI 572 in Spring 2015 and with Nutch PMC member Mo Omer) that will allow your crawler to interact with any dynamic Ajax based content, and to access and enter the "Deep Web" to get at the weapons data available from these sites. There are several additional objectives including finding as much content as possible, but doing so in such a way that your team adheres to all of the things we have discussed about to date in class related to crawling, e.g., politeness, parallel crawling, all the while identifying the expanse of the web graph and the portion of the domain and graph that you are crawling. You are also required to ensure that you are not simply crawling and acquiring meaningless content on these sites – rest assured the goal is *not* for you to download all the javascript and CSS files as part of crawling these sites.

While crawling and performing the above objective, you will run into a number of issues ranging from having to negotiate different protocols to get to the actual weapons images – some are behind FTP, HTTP, HTTPS, etc.; to navigation requiring web forms and having to POST to those forms to navigate to the actual content. Some issues will relate to having to execute Javascript even, and even interpret the page in order to obtain content from it – a technique that GoogleBot and other crawlers are actively adopting as we discussed in class. So think carefully about your work and plan before you jump head first into crawling. How are you going to deal with *politeness*? How are you going to deal with *completeness*? How are you going to deal with *dark data* and the web that exists behind *forms*? And finally how are you going to deal with *dynamic content*?

You will use Nutch as the core framework to perform crawling, and Tika as the main content detection and extraction framework. More information (installing Nutch, and Tika, etc.) will be provided in Section 4. In addition, you are going to use some in-development plugins and code for Nutch and Tika that will help you understand what is required to properly deal with extraction of information from the content. You will need to extract and understand text, metadata, and language from the content in order to develop an algorithm to perform deduplication. You are responsible for constructing two algorithm types for deduplication – one that identifies *exact* matches, and another for *near duplicates*. This will be discussed in more detail below.

The assignment specific tasks will be specified in the following section.

3. Tasks

1. Download and configure Nutch to crawl Weapons images as identified in the seed list that will be sent to you by the graders
    a. Identify and make changes to Nutch configuration to deal with politeness.
        i. Hint this may involve using the rotating Agent ID plugin, and setting many of the configurable properties in the nutch-default.xml or nutch-site.xml file and also consider information on this page: http://wiki.apache.org/nutch/WhiteListRobots

   b. Identify and make changes to Nutch configuration to deal with URL filtering.
   c. Create team identification information and label your Nutch bot via Nutch configuration.
2. Perform crawls of Weapons images sites
   a. Use the NutchPy library from http://github.com/ContinumIO/nutchpy
   b. Use NutchPy to read Nutch crawl data and write a program or script to classify what image MIME types you encounter
   c. Identify at least 10 different image MIME types that you encounter while crawling.
   d. Deliver a list of at least 100 URLs that you have difficulty fetching and identify why (e.g., protocol issue, behind a web form, requires Ajax, etc.).
   e. Deliver the crawl statistics from your crawls of each repository.
      i. Crawl statistics should be information e.g., about what URLs crawled, the HTTP response, etc. You can deliver this as a text file included with your final deliverable zip outlined in Section 6.
3. Use the information you learn in Part 2 to extend the Nutch Selenium Protocol Plugin
   a. See this wiki page https://wiki.apache.org/nutch/AdvancedAjaxInteraction
   b. Extend and develop a plugin for protocol-interactive-selenium to better handle the problem URLs you have identified in step 2
   c. Rebuild Nutch with your plugin, and enable your plugin.
4. Build the latest version of Tika and Install Tesseract
   a. Install Tesseract and Tika via https://wiki.apache.org/tika/TikaOCR .
   b. Download and install Nutch trunk.
   c. Upgrade Tika in Nutch trunk with http://svn.apache.org/repos/asf/nutch/trunk/src/plugin/parse-tika/howto_upgrade_tika.txt .
5. Re-run your Weapons crawls with the enhanced Tika and Nutch Selenium you've built in step 3
   a. Identify at least 100 URLs that you have difficulty fetching and identify why
   b. Are the URLs present from 2d still?
   c. Did the enhanced Tika parsing assist with that?
   d. Deliver your updated crawl statistics from each repository your crawl.
6. Develop two deduplication algorithms to use on the extracted text and metadata in the Parsed Content from Nutch
   a. One for exact duplicates using the Weapons data and text and metadata from your crawls.
   b. One for near duplicates using the Weapons data and text and metadata from your crawls.
7. Enable the Nutch Similarity Scoring Filter Focused Crawling Plugin
   a. See documentation here: https://wiki.apache.org/nutch/SimilarityScoringFilter
   b. Re-run your Weapons crawls using this Filter

    c. Compare the resultant URLs from using this Filter to your results from Step 6 – were your deduplication algorithms more effective? Which ones (6a or 6b) and why?
    d. Deliver your updated crawl statistics using the Similarity Scoring Filter.
8. (**EXTRA CREDIT**) Download and configure Nutch Python, which you will use to control your crawls
    a. You can find Nutch Python here (http://github.com/chrismattmann/nutch-python/)
    b. Write a Python program that will perform your crawls with Nutch using the Nutch REST server and nutch-python
    c. See documentation here: https://wiki.apache.org/nutch/Nutch_1.X_RESTAPI
    d. Consider how to take the Java code in NutchPy (ContinuumIO's plugin) and how they could be made Nutch REST services.
9. **(EXTRA CREDIT)** Dump the crawled data out of your Nutch content and then run https://github.com/chrismattmann/tika-similarity/ over it.
    a. Take the resultant JSON output and visualize it using D3 per the in instructions in the Github repo.
    b. Are there any interesting clusters?
    c. Can you explain the clusters that you see?
10. **(EXTRA CREDIT)** Re-run your crawls using Memex Explorer, a Domain Specific Search Tool, available at: http://github.com/memex-explorer/memex-explorer
    a. Describe any bugs you encountered in Memex Explorer
    b. Were you able to run your crawls?
    c. What was missing from Memex Explorer's Nutch capabilities?

4. Assignment Setup

4.1 Group Formation

You can work on this assignment in groups sized 4. Komal, the lead grader, will send you instructions on how to formulate groups. If you have questions contact:

Mohit Bagde
bagde@usc.edu

Divydeep Agarwal
divydeea@usc.edu

Komal Dhawan
komaldha@usc.edu

Use subject: CS 572: Team Details

4.2 Seed URLs for Weapons

The seed URLs for weapons will be provided to each group by the Graders. Once groups are formed, team leads should request the seed URL list from the grader emails above. Please send **one** email requesting the seed URLs per team.

You should begin collecting data as soon as you can since you will likely have to recrawl many times to get this right.

4.3 Downloading Apache Nutch

To get started with Apache Nutch, grab the trunk version, available from:

http://svn.apache.org/repos/asf/nutch/trunk
http://github.com/apache/nutch/

The latest version of Nutch that we will use in this assignment is Nutch 1.11-trunk.
Build Nutch with Apache Ant (available on most Unix systems) and Ivy by first checking out the code using a Subversion client:

```
svn co http://svn.apache.org/repos/asf/nutch/trunk nutch
```

or via Github:

```
git clone http://github.com/apache/nutch.git
```

Then enter the nutch directory and type:

```
ant runtime
```

Then wait a while. After Nutch is done building, you will have a runtime/local directory. That will be the Nutch deployment. The directory looks like this:

```
bin  conf  lib  logs  plugins  test
```

4.4 Downloading and Installing Apache Tika

The quickest and best way to get Apache Tika up and running on your machine is to grab the `tika-app.jar` from: http://tika.apache.org/download.html. You should obtain a jar file called `tika-app-1.10.jar`. This jar contains all of the necessary dependencies to get up and running with Tika by calling it your Java program.

Documentation is available on the Apache Tika webpage at http://tika.apache.org/. API documentation can be found at http://tika.apache.org/1.10/api/.

You can also get more information about Tika by checking out the book written by Professor Mattmann called "Tika in Action", available from: http://manning.com/mattmann/.

4.5 Some hints and helpful URLs

You may want to look into the Nutchpy library from Continuum Analytics as a method for generating crawling statistics from your Nutch databases and for your program to classify MIME types (task 2b):

https://github.com/ContinuumIO/nutchpy

A discussion of using Nutch to crawl dark data behind Ajax is ongoing and relevant to the Selenium task 4:

https://wiki.apache.org/nutch/AdvancedAjaxInteraction

Since you will be working with Interactive Selenium, you may want to look up this page on the Selenium plugin in Nutch:
https://github.com/apache/nutch/tree/trunk/src/plugin/protocol-interactiveselenium

If you do the extra credit, you will need to look at D3:

http://d3js.org/

5. Report

Write a short 4 page report describing your observations, i.e. what you noticed about the dataset as you completed the tasks. Why do you think there were duplicates? Were they easy to detect? Describe your algorithms for deduplication. How did you arrive at it? What worked about it? What didn't? Describe the URLs that worked and why? What MIME types did you retrieve from these websites?

Thinking more broadly, do you have enough information to answer the following:

Was there a particular correlation between gun type and website?

Were duplicates indicative of sloppy selling, or simply dealers promulgating the product to multiple sites?

Are there any weapons that aren't necessarily firearms, but more dangerous devices (e.g., explosives)?

What relationships do the clusters resultant from your deduplication algorithms tell you? Are they simply related to the ways that the pictures are edited, or indicative of anything more?

Also include your thoughts about Apache Nutch and Apache Tika – what was easy about using them? What wasn't?

6. Submission Guidelines

This assignment is to be submitted ***electronically, by 12pm PT*** on the specified due date, via Gmail csci572fall2015@gmail.com. Use the subject line: CSCI 572: Mattmann: Fall 2015: Nutch Homework: <Your Lastname>: <Your Firstname>. So if your name was Lord Voldemort, you would submit an email to csci572fall2015@gmail.com with the subject "CSCI 572: Mattmann: Fall 2015: Nutch Homework: Voldemort: Lord" (no quotes). **Please note only one submission per team**.

- All source code is expected to be commented, to compile, and to run. You should have (at least) one Java source file for part 3 and extending the interactive selenium plugin and you should also include other java source files that you added, if any. Do **not** submit *.class files. We will compile your program from submitted source.
- Include your program and/or script for generating MIME stats from a Nutch database. You must use Python (NutchPy) to generate this.
- Deliver your Nutch configuration e.g., your nutch-default.xml, nutch-site.xml, urlfilter-regex.txt file and any other configuration necessary to reproduce your results.
- Teams will be asked if they would like to contribute their crawled dataset to our DARPA Memex Amazon machine. This would include your Nutch segment data for each repository. If you want to do this, please identify in your report that you would like to do this, and send a message to the professor, to the TAs, and to the graders.
- Also prepare a readme.txt containing any notes you'd like to submit.
- Do **not** include tika-app-1.10.jar in your submission. We already have this.
- However, if you have used any external libraries other than Tika, you should include those jar files in your submission, and include in your readme.txt a detailed explanation of how to use these libraries when compiling and executing your program.
- Save your report as a PDF file (Lastname_Firstname_NUTCH.pdf) and include it in your submission.
- Compress all of the above into a single zip archive and name it according to the following filename convention:

  <lastname>_<firstname>_CSCI572_HW_NUTCH.zip

  Use only standard zip format. Do **not** use other formats such as zipx, rar, ace, etc.
- If your homework submission exceeds the Gmail's 25MB limit, upload the zip file to Google drive and share it with csci572fall2015@gmail.com

Important Note:

- Make sure that you have attached the file the when submitting. Failure to do so will be treated as non-submission.
- Successful submission will be indicated in the assignment's submission history. We advise that you check to verify the timestamp, download and double check your zip file for good measure.
- Again, please note, only **one submission per team**. Designate someone to submit.

6.1 Late Assignment Policy

- -10% if submitted within the first 24 hours
- -15% for each additional 24 hours or part thereof